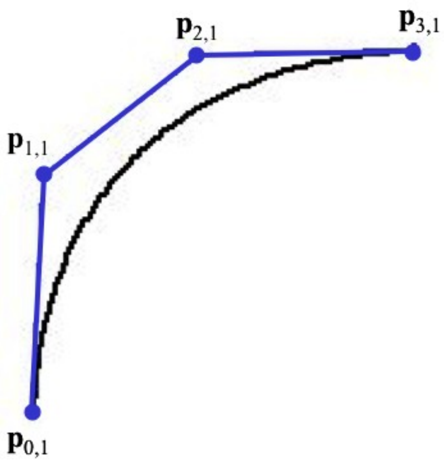
Übungsblatt 11

Willkommen zur elften Übung der Veranstaltung *Generative Computergrafik*. Ziel dieses Übungsblatts ist, dass Sie sich mit *rationalen B-Spline-Kurven* vertraut machen. Aufgabe 3 ist bis zum 10. Juli 2020 um 23:55 Uhr über `https://read.mi.hs-rm.de` abzugeben.

Aufgabe 1. Gegeben sei die unten abgebildete kubische Bézier-Kurve

$$\mathbf{x}_1(t) = \sum_{i=0}^{3} B^3_{i_{[1,2]}}(t)\mathbf{p}_{i,1}$$

über dem Parameterintervall $[1, 2]$. Diese sei Bestandteil einer kubischen Bézier-Splinekurve mit den Parameterintervallen $[0.5, 1]$, $[1, 2]$, und $[2, 3]$.



1. Konstruieren Sie eine Bézier-Kurve $\mathbf{x}_2(t)$ (über $[2, 3]$), welche im Punkt $\mathbf{p}_{3,1}$ zweimal stetig differenzierbar an die Kurve $\mathbf{x}_1$ anschließt.
2. Konstruieren Sie eine Bézier-Kurve $\mathbf{x}_0(t)$ (über $[0.5, 1]$), welche im Punkt $\mathbf{p}_{0,1}$ zweimal stetig differenzierbar an die Kurve $\mathbf{x}_1$ anschließt.

Aufgabe 2. Gegeben sei die B-Spline-Kurve $\mathbf{x}(t) = \sum_{i=0}^{3} N_i^3(t)\mathbf{b}_i$ der Ordnung 3 (vom Polynomgrad 2) mit Knotenvektor $T = (0, 0, 0, 1, 2, 2, 2)$ und den Kontrollpunkten $\mathbf{b}_0 = (0, 0)^T, \mathbf{b}_1 = (1, 2)^T, \mathbf{b}_2 = (5, 2)^T, \mathbf{b}_3 = (6, 0)^T$. Ermitteln Sie die Kurvenpunkte $\mathbf{x}(1/2)$ und $\mathbf{x}(3/2)$.

Aufgabe 3. Schreiben Sie ein `glfw`-Programm (**verwenden Sie <u>nicht</u> das mit Übungsblatt 10 vorgegebene** `TKINTER`**-Skript** `BEZIERTEMPLATE.PY`), welches analog zu Aufgabe 2 auf Übungsblatt 10, die Eingabe von Punkten $\mathbf{p}_0, \ldots, \mathbf{p}_n$ mit Hilfe der Maus erlaubt und das Polygon durch die Punkte sowie eine Kurve zeichnet. Die dargestellte Kurve soll eine **B-Spline-Kurve**

$$\mathbf{x}(t) = \sum_{i=0}^{n} N_i^k(t)\mathbf{p}_i$$

der Ordnung k mit uniformen Knotenvektor (der Länge $n + k + 1$)

$$K = \{\underbrace{0, \ldots, 0}_{k}, 1, 2, \ldots, n-(k-1), \underbrace{n-(k-2), \ldots, n-(k-2)}_{k}\}$$

und (Kontroll-)Punkten $\mathbf{p}_0, \ldots, \mathbf{p}_n$ sein.

1. Gehen Sie bei der Implementierung analog zu Teil 1 von Aufgabe 2 auf Blatt 10 vor. Schreiben Sie dazu eine Funktion

   ```
   deboor(degree, controlpoints, knotvector, t)
   ```

   die mit Hilfe des *de-Boor-Algorithmus* einen Punkt auf einer B-Spline-Kurve berechnet. Die einzelnen Parameter der Funktion haben dabei folgende Bedeutungen:

   - `degree`: **Polynomgrad** der B-Spline-Kurve
   - `controllpoints`: **Kontrollpunkte** der B-Spline-Kurve
   - `knotvector`: **Knotenvektor** der B-Spline-Kurve
   - `t`: **Parameterwert**, zu dem der Punkt auf der B-Spline-Kurve berechnet werden soll

   Verwenden Sie die Funktion `deboor()` dann, um eine feste Anzahl m von Punkten auf der Kurve zu berechnen und anschließend als Polygon darzustellen. Beachten Sie, dass eine B-Spline-Kurve der Ordnung k erst dann vollständig definiert ist, wenn wenigstens k Punkte angegeben wurden. Bis zur Eingabe des k-ten Punktes soll lediglich das (Kontroll-)Polygon durch die Punkte dargestellt werden.

   Ihr Programm soll es weiterhin ermöglichen die beiden Parameter k (*Ordnung* der Kurve) und m (*Anzahl* zu berechnender *Kurvenpunkte)* zu beeinflussen. Die *Ordnung* soll dabei mit den Tasten `k` bzw. `K`, die *Anzahl Kurvenpunkte* mit den Tasten `m` bzw. `M` *verringert* bzw. *erhöht* werden können.

2. Erweitern Sie Ihr Programm aus Teil 1. so dass es eine **nicht uniforme rationale B-Spline-Kurve (NURBS)**

$$\mathbf{x}(t) = \sum_{i=0}^{n} N_i^k(t)\tilde{\mathbf{p}}_i$$

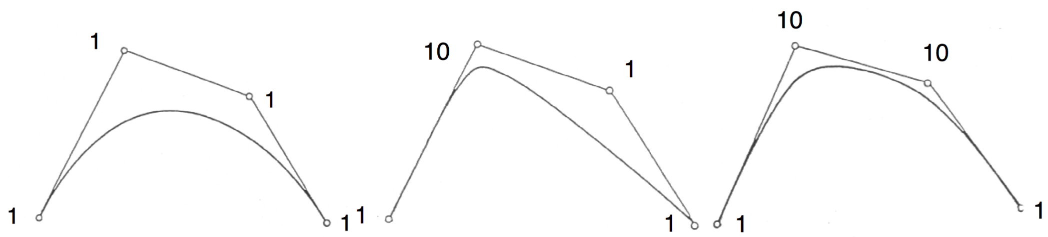
der Ordnung k mit Knotenvektor (der Länge $n + k + 1$)

$$K = \{\underbrace{0, \ldots, 0}_{k}, 1, 2, \ldots, n - (k-1), \underbrace{n - (k-2), \ldots, n - (k-2)}_{k}\}$$

und *homogenen (Kontroll-)Punkten* $\tilde{\mathbf{p}}_0 = \omega_0(x_0, y_0, 1)^\top, \ldots, \tilde{\mathbf{p}}_n = \omega_n(x_n, y_n, 1)^\top$ darstellen kann. Die Gewichte ω_i können als zusätzliche Designparameter eingesetzt werden.

Ihr Programm soll zusätzlich zur Funktionalität aus Teil 1. die Möglichkeit bieten, das Gewicht ω_i jedes Kontrollpunkts $\mathbf{p}_i$ im Intervall $[1, 10]$ mit der Maus zu ändern (z.B. indem man bei gedrückter SHIFT-Taste auf den entsprechenden Kontrollpunkt klickt und die Maus bewegt).

Die nachfolgenden Abbildungen zeigen, wie sich die Kurven bei Veränderung der entsprechenden Gewichte verhält.



Haben alle Gewichte den gleichen Wert (z.B. $\omega_i = 1 \forall i$) so handelt es sich bei der rationalen Kurve um eine einfache polynomielle Kurve (siehe Abbildung links). Wird das Gewicht eines Kontrollpunktes (relativ zu den Gewichten der benachbarten Kontrollpunkte) erhöht, so wird die Kurve zu dem entsprechenden Kontrollpunkt hingezogen (siehe mittlere und rechte Abbildung).